## Transpose Matrix

Sheldon Cooper has a square-shaped puzzle made up of small square pieces containing numbers on them. He wants to rearrange the puzzle by changing the elements of a row into a column element and the column element into a row element without altering the shape of the puzzle. Help Sheldon solves this puzzle. Write a program to find the transpose of the given 2D matrix.

**Input Format**

The first line consists of an integer which represents the number of rows and columns. The next line consists of the elements in the matrix.

**Output Format**

Output prints the transpose of the input matrix.

**Sample Input 0**

3

1 2 3

4 5 6

7 8 9

**Sample Output 0**

1 2 3

4 5 6

7 8 9

Transpose matrix is:

1 4 7

2 5 8

3 6 9

//SOURCE CODE

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int ar[][]=new int[n][n];

for(int i=0;i<n;i++)

{

for(int j=0;j<n;j++){

ar[i][j]=sc.nextInt();

}

}

for(int i=0;i<n;i++){

for(int j=0;j<n;j++){

System.out.print(ar[i][j]+" ");

}

System.out.println();

}

System.out.println("Transpose matrix is:");

for(int i=0;i<n;i++)

{

for(int j=0;j<n;j++)

{

System.out.print(ar[j][i]+" ");

}

System.out.println();

}

}

}

## Matrix Multiplication

Write a program to multiply two matrices of the same dimensions together.

**Input Format**

The first two integer inputs, m, and n represent the dimensions of both of the matrices that have to be multiplied. Next (m\*n) integers represent the elements in the first matrix and followed by another (m\*n) integer representing the elements in the second matrix

**Output Format**

The output consists of (m\*n) integers representing the values of the matrix that is the product of both the input matrices.

**Sample Input 0**

3 3

1 2 3

4 5 6

7 8 9

9 8 7

6 5 4

3 2 1

**Sample Output 0**

30 24 18

84 69 54

138 114 90

//SOURCE CODE

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int m=sc.nextInt();

int f=sc.nextInt();

int ar[][]=new int[m][f];

for(int i=0;i<m;i++)

{

for(int j=0;j<f;j++)

{

ar[i][j]=sc.nextInt();

}

}

int ar1[][]=new int[m][f];

for(int i=0;i<m;i++)

{

for(int j=0;j<f;j++)

{

ar1[i][j]=sc.nextInt();

}

}

int ar2[][]=new int[m][f];

for(int i=0;i<m;i++)

{

for(int j=0;j<f;j++)

{

ar2[i][j]=0;

for(int k=0;k<m;k++){

ar2[i][j]+=ar[i][k]\*ar1[k][j];

}

System.out.print(ar2[i][j]+" ");

}

System.out.println();

}

}

}

## Sum of Zig-Zag

Ravi is not able to figure out the method to calculate the sum of the Zig-Zag pattern in the Matrix. Can you help Ravi to write a program to find the sum of Zig-Zag patterns in a given matrix?

**Input Format**

* Input consists of 2 integers and 1 2D array.
* Integers corresponding to the size of rows and columns.

**Output Format**

* The output prints the sum of the zig-zag pattern.
* Refer to the sample output.
* Note: Size of row and column should be same

**Sample Input 0**

3

3

1 2 3

4 5 6

7 8 9

**Sample Output 0**

Sum of Zig-Zag pattern is 35

//SOURCE CODE

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int m=sc.nextInt();

int f=sc.nextInt();

int ar[][]=new int[m][f];

for(int i=0;i<m;i++)

{

for(int j=0;j<f;j++)

{

ar[i][j]=sc.nextInt();

}

}

int sum=0;

int temp=f;

for(int i=0;i<m;i++)

{

if (i==0||i==m-1)

{

for(int j=0;j<f;j++)

{

sum+=ar[i][j];

}

}

else {

for(int j=0;j<f;j++)

{

if((i+j)==(m-1))

{

sum+=ar[i][j];

}

}

}

}

System.out.print("Sum of Zig-Zag pattern is "+sum);

}

}